

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Dalam pengiriman terhadap pengembang bisnis, awalnya B-Log hanya dapat mengantarkan pesanan terhadap satu konsumen dengan cara konsumen sudah melakukan persetujuan dengan B-Log, lalu melakukan pemesanan dengan mengirimkan melalui *platform* Webmail dengan berbentuk *file* Excel, lalu *file* yang dikirimkan tadi akan dimasukkan oleh admin kedalam aplikasi pengantaran B-Log. Pengantaran akan dilakukan ketika admin mengunduh *template* yang terdapat pada aplikasi tersebut, lalu mengunggah *file* tersebut, *file* tadi akan di verifikasi terlebih dahulu sebelum masuk ke dalam *database*. Apabila ada data yang salah, maka akan dilakukan *edit*, dan kalau data nya sudah benar akan dilakukan penyimpanan (*save*). Lalu seiring berjalannya waktu, tingkat pesanan dari customer meningkat sehingga B-Log memutuskan untuk membuat *module* yang bernama Order Management System. OMS merupakan *module* yang bisa digunakan B-Log untuk dapat melayani pengembangan bisnis yang terjadi. Pengembang OMS menggunakan *micro services* dimana untuk membuat aplikasi menjadi lebih ringkas. OMS juga dapat menambahkan jumlah konsumen dalam setiap pengantaran sehingga dalam melakukan pengantaran dapat lebih cepat. Salah satu hal yang dibutuhkan pada pembuatan *module* OMS adalah API (*Application programming interface*) yang dibutuhkan untuk menjadi jembatan antara *database* dengan tampilan *website*. *Application Programming Interface* (API) adalah sebuah antarmuka yang dibangun oleh pengembang sistem informasi sehingga beberapa fungsi atau seluruh fungsi dari sistem dapat diakses secara terukur atau terprogram[1]. Dalam komputasi, API merupakan antarmuka yang memungkinkan aplikasi perangkat lunak atau kombinasi perangkat keras dan perangkat lunak untuk saling berkomunikasi dan bekerja sama[2]. Dengan adanya API, untuk menampilkan data atau menerima data tidak lagi sulit karena dengan adanya API hanya perlu melakukan request terhadap API yang diinginkan dan tidak perlu mencari tabel dan kolom *database* yang rumit[2].

Pelaksanaan kerja magang kali ini dilakukan dengan menggunakan bahasa pemrograman Go-Lang dan menggunakan *framework* GORM (Go Object Relation Mapping), dan Gin dengan menggunakan *database* Postgresql. Hal ini dikarenakan

kinerja tinggi yang sangat responsif, cocok untuk menangani banyak permintaan sekaligus tanpa terhambat. Selain itu, fitur konkurensi dan paralelisme di Go memudahkan pengembang dalam membuat kode yang efisien untuk menangani banyak proses secara bersamaan. Sintaksisnya juga sederhana, dan sudah ada banyak pustaka bawaan yang mempermudah pengembangan tanpa harus mengandalkan pustaka eksternal. Go juga memudahkan proses *deployment* dan pengujian dengan kemampuan menghasilkan satu binary mandiri dan dukungan bawaan untuk pengujian. Keunggulan lainnya adalah skalabilitas dan stabilitas yang baik, didukung oleh komunitas pengembang yang aktif[3].

GORM merupakan ORM (Object-Relational Mapping) yang kuat dan mudah digunakan, memungkinkan pengembang untuk berinteraksi dengan *database* secara efisien menggunakan bahasa pemrograman Go tanpa harus menulis SQL secara langsung. Ini mempercepat proses pengembangan dan meminimalkan kemungkinan kesalahan dalam kueri *database*[4].

Gin merupakan *framework website* yang ringan dan cepat, dirancang khusus untuk pengembangan API dengan kinerja tinggi. Dibandingkan dengan beberapa *framework* lainnya, Gin memiliki overhead yang rendah, membuatnya cocok untuk menangani beban tinggi dan memproses permintaan HTTP dengan cepat. Gin juga menyediakan fitur *routing* yang kuat dan mudah digunakan, serta dukungan untuk *middleware*, memungkinkan pengembang untuk menambahkan fungsionalitas tambahan seperti autentikasi, *logging*, dan manajemen kesalahan dengan mudah[5].

PostgreSQL menonjol sebagai sistem manajemen basis data yang andal dan kuat dengan dukungan untuk berbagai jenis data dan beban kerja. Dikenal karena kinerja yang baik bahkan dalam situasi kompleks, PostgreSQL menawarkan fitur-fitur seperti *optimizer query* yang canggih, keamanan tinggi, dan dukungan komunitas yang besar. Dengan kombinasi fleksibilitas, kinerja, dan keamanan, PostgreSQL menjadi pilihan yang sangat baik untuk berbagai aplikasi *database*[6].

1.2 Maksud dan Tujuan Kerja Magang

Maksud dari pelaksanaan kerja magang adalah sebagai berikut,

1. Meningkatkan *hard skill* dengan memperdalam bahasa pemrograman Go dan *soft skill* dengan berkolaborasi dengan rekan kerja.
2. Mendapatkan pengalaman pada dunia kerja dengan keterlibatan sebagai

karyawan terhadap suatu perusahaan.

3. Melakukan implementasi terhadap ilmu yang sudah di dapatkan dalam dunia kerja.

Pelaksanaan kerja magang sebagai *website backend developer* bertujuan untuk membuat API yang akan digunakan pada *module* OMS (Order Management System)

1.3 Waktu dan Prosedur Pelaksanaan Kerja Magang

Waktu pelaksanaan kerja magang secara kontrak tertulis dimulai dari 21 Maret 2024 dan berakhir pada 20 September 2024 dengan minimum waktu yang dicapai adalah 640 jam. Dalam laporan yang ditulis pelaksanaan dimulai dari 20 Februari. Pelaksanaan kerja magang dibimbing oleh pembimbing lapangan, yaitu Hariesa Budi Prabowo selaku *IT Department Head*. Pelaksanaan kerja magang dilakukan sebagai berikut :

1. Pelaksanaan kerja magang dilakukan dari hari Senin - Jumat dari pukul 08:00 WIB - 17:00 WIB.
2. Waktu kerja magang setiap harinya adalah 8 jam dengan total 40 jam setiap minggu nya.
3. Pelaksanaan kerja magang dilakukan secara *WFO (Work From Office)* dengan langsung datang ke kantor setiap harinya.

U M M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A